MFPT calculation for random walks in inhomogeneous networks

Isuri Wijesundera

Department of Infrastructure Engineering, The University of Melbourne, Australia

Malka N. Halgamuge, Ampalavanapillai Nirmalathas

Department of Electrical and Electronic Engineering, The University of Melbourne

Thrishantha Nanayakkara

Division of Informatics, King's College London, London, UK

Abstract

Knowing the expected arrival time at a particular state, also known as the mean first passage time (*MFPT*), often plays an important role for a large class of random walkers in their respective state-spaces. Contrasting to ideal conditions required by recent advancements on *MFPT* estimations, many naturally occurring random walkers encounter inhomogeneity of transport characteristics in the networks they walk on. This paper presents a heuristic method to divide an inhomogeneous network into homogeneous network primitives (NPs) optimized using particle swarm optimizer, and to use a 'hop-wise' MFPT calculation method. This methodology's potential is demonstrated through simulated random walks and with a case study using the dataset of past cyclone tracks over the North Atlantic ocean. Parallel processing was used to increase calculation efficiency. The predictions using the proposed method is compared to real data averages and predictions assuming homogeneous transport properties. The results show that breaking the problem into NPs reduces the average error from 18.8% to 5.4% with respect to the homogeneous network assumption.

Keywords: MFPT, random walk, inhomogeneous, network primitives, PSO,

Email address: isuriw@student.unimelb.edu.au (Isuri Wijesundera)

1. Introduction

The significant role played by first encounters have made calculating the mean first passage time (MFPT) of great importance in numerous real world dynamic systems that can be modeled as random walks [1, 2]. From exchange rate fluctuations [3], through natural disaster propagation [4, 5], to the propagation of gossip [6], the MFPT to reach some special target state gives important information of the performance of the system. For this reason, random walks have been studied extensively for decades on one dimensional state space [7, 8], regular lattices, fractal networks, and many other specialized networks [9]. Current work on MFPT prediction has advanced to finding efficient and accurate methods for random walks on a range of networks including those on complex scale invariant media [10], fractals [11], self-similar networks [12, 13], scale-free networks [14, 15], and as lévy flights [16, 10].

Many prediction models have considered network nodes as *well-mixed homogeneous populations* [17, 18, 19, 6]. Although many methods produce excellent results on estimating MFPT on different types of networks, a common requirement is that all nodes in the network are homogeneous in some transport properties that are used in MFPT calculations. In other words, the properties of the network and the random walker need to be length-scale invariant [20]. Some examples of transport properties used in MFPT calculations are the spectral dimension (ds [21]) as used in [22], and fractal dimension (df) and walk dimension (dw) as used in [10]. Therefore to use these methods, the transport properties used in MFPT calculations need to be the same at all parts of the network.

However, networks showing strong spatial variability in transport properties have been proven to represent a considerable portion of real world systems; both man-made and natural [9]. Random walks are among the basic stochastic processes which are affected by inhomogeneity of a network. Some simple real

world examples of inhomogeneous networks are the atmosphere on which a cyclone would travel through, a terrain on which flood will propagate through, an area where fire will spread through, the density distribution of people through which a disease will spread or even the path a piece of gossip will travel. Further applications are found where the state of a dynamic system is expressed in terms of dimensions apart from the Cartesian spatial changes as per the above examples. For example, if the discrete state transitions of a rimless wheel rolling down a ramp is observed with respect to the temporal value of its angular velocity, a random walk can be observed on the return-map-space of angular velocity [23] which is inhomogeneous since the statistics of visiting nodes are not evenly distributed.

In the real world, the networks representing the mobility pattern of individuals among different subpopulations are in many cases highly inhomogeneous [17]. The authors of [24] have shown that inhomogeneous networks are more vulnerable to iterative path attacks and have shown in [25] how epidemic spreading is dependent on network homogeneity. In [26] and [27], network capacity is shown to decrease with network inhomogeneity. In animal motion it has been shown that the calculation for expected time needed for a predator to locate small patches of prey in a 2-D landscape has two components; random and directed [28]. Although animal motion itself is isotropic but possibly spatially inhomogeneous due to inhomogeneity in landscape. Random walks in all these scenarios will go through different portions of a network where the network itself will behave in such different ways making the estimation of arrival time at a given destination extremely difficult to calculate as a whole.

Network inhomogeneity could arise from the mere design the network was made for. Nevertheless, this inhomogeneity affects the many stochastic processes which occur on the network such as random walks. Although network inhomogeneity generally increases the network analysis and design complexity, many advantages have also been identified. The authors in [29] show that a spatially inhomogeneous search can indeed significantly enhance the rate of arrival at the target. They have shown that in optimal inhomogeneous search

Table 1: List of abbreviations

AIC	Akaike Information Criterion
BIC	Bayesian Information Criterion
FPT	First Passage Time
GMM	Gaussian Mixture Model
GUI	Graphical User Interface
HPSO	self-organizing Hierarchical Particle Swarm Optimizer
MAP	Maximum A'Posteriori
MFPT	Mean First Passage Time
NA	North Atlantic
NP	Network Primitive
PSO	Particle Swarm Optimization
SF	Scale Free

tests, MFPT on such networks are completely dominated by those trajectories heading directly towards the target.

Network inhomogeneity has been defined in various ways in literature. The most common definition is the inhomogeneity in terms of node degree distribution [30, 31, 9, 26]. One of the most common such network type that is studied in literature is the scale free (SF) networks [32] which show a power-law degree distribution [9, 24]. Since in this paper we focus on developing generic methods to estimate MFPT of random walks, we define network inhomogeneity as the spatial inhomogeneity in transport properties (e.g. *df* as a metric of node density, *dw* as a metric of the ease of propagation) experienced by a random walker on that network.

1.1. Current trends in MFPT estimation in inhomogeneous networks

Much effort has been made on analyzing basic properties of random walks on inhomogeneous networks such as small-world type [18] and SF networks [31]. The findings in [33] show that the MFPT depends on source-target distance and

Table 2: Symbols, variables and notations

H	Network primitive type (Hypothesis) distribution
$\boldsymbol{N}$	Number of particles for optimization
N	Number of nodes in network
A, B	Domain dependent constants
df	Fractal dimension
ds	Spectral dimension
dw	Walk dimension
dw_b	Bias modified walk dimension
M	Number of ellipses
n	Number of hops in hop-wise estimation
$\boldsymbol{n}$	Total number of dimensions
r	Distance
R	Rate of spread
S	Source
T	Target
T_i	MFPT for i^{th} hop
P_{DD}	Probability density distribution
$MFPT$	Mean first passage time
η	Number of NP types
θ	Bias direction
$\boldsymbol{N}(m, \sigma)$	Normal distribution with m mean and σ standard deviation
$\mathcal{O}(f)$	Big-O complexity of function f

the degree-distribution for SF networks. They have also shown significant differences of these properties when the networks show non-compact explorations as opposed to compact explorations [34]. The transit and commute times tend to diverge when the network consists of more than one cluster because the graphs are not connected according to [31] where a method to estimate MFPT on Erdös-Renyi random graph has been presented.

Most of these methods which address the question of estimating MFPT for random walks on inhomogeneous networks provide a global result which is the average of MFPT over a set of starting points distributed uniformly over all the other nodes of the graph [29, 9]. And a large focus has been set on SF networks [29]. The reason being that there are a considerable class of networks which satisfy the SF condition. However, the focus of this paper is on developing generic methods of estimating the MFPT for random walks on networks that are inhomogeneous in transport properties and which are not necessarily SF.

Using the knowledge that many inhomogeneous networks commonly include frequent homogeneous patches (e.g. fuel distributions for fire propagation, distribution of cities for disease propagation, etc.), it is understood that if it is possible to divide the network into portions where all nodes within that sub-network have (approximately) similar transport properties for MFPT calculations, the complexity of the problem can reduce considerably as MFPT prediction between any two points within such network portion would be straight forward using methods such as that presented in [10]. In our previous work in [35, 36] we have presented a method of reducing the calculation complexity for inhomogeneous networks by dividing the problem of predicting flood propagation through identifying 'Geographic Primitives' based on the terrain slope profile. In this paper, a somewhat different approach is taken following a more generic method to identify portions of a network with homogeneous transport properties of a random walk on that network. Then we present a method to calculate the MFPT between any source and target pair using a 'hop-wise' calculation approach and using an extension of a powerful method from [10] to calculate MFPT for each 'hop'. The next section discusses methods of partitioning an

inhomogeneous network for the purpose of MFPT calculation.

1.2. Network partitioning

At this point the importance in dividing the problem into simpler portions was identified: that is to follow a divide-and-conquer method to reduce the complexity of the problem. The next question is which properties of a network should be considered when trying to divide the network. There were numerous ways in which the network could be divided by. The work in [29] has considered the network as two concentric regions with piece-wise constant diffusivity. Ferguson et. al. has shown in [37] many ways in which an inhomogeneous network can be looked at in terms of epidemic transmission. The first and the most basic method is to assume homogeneous mixing. However, this suppresses many features of the transmission and does not give adequate intelligence of the system. The first method of network division proposed was by age/social structure to account for different probabilities of contact between these subgroups. The next was to identify a static network structure. But this has been shown to be difficult if not impossible for many systems [18]. The fourth method is to identify homogeneous patches in the network. Such patch, or a subpopulation is generally a natural unit of study. Although deterministic models which give exact solutions provide rapid simulations it is extremely important to account for stochasticity in transmission events [37].

This paper follows the patch identification method to divide the network with the requirement that patches remain homogeneous. We propose that if we divide the network into patches homogeneous in the transport properties that are used to calculate MFPT within a patch, that would be a viable solution for calculating MFPT for random walks in networks that are inhomogeneous in transport properties. A hop-wise approach is followed in obtaining the MFPT for any predefined source-target set on an inhomogeneous network divided into homogeneous primitives. Multiple possible paths depending on patch exit points, are weighed to get the final path-integral representation of random walk properties.

The remainder of this paper is as follows. In Section 2.1 the concept of Network Primitives is introduced for the problem of estimating MFPT for random walks on an inhomogeneous network. Section 2.2 outlines the MFPT prediction method used *within* an identified NP. Section 2.3 extends and combines the NP identification process with the MFPT calculation methods to facilitate the MFPT estimation process as a hop-wise calculation. Section 2.4 details the developed NP identification process with an example generic discrete 2D network and Section 2.5 presents a case study using a dataset of cyclone tracks recorded above the North Atlantic Ocean. Finally, a comparison of results with real data averages and with estimations assuming transport variable homogeneity is presented in Section 3 and Section 4 concludes the paper. All abbreviations and acronyms used in this paper are summarized in Table 1 and the common notation, symbols and variables are presented in Table 2.

2. Methodology

2.1. Network Primitives (NPs)

In the context of transport property based MFPT calculations for an inhomogeneous node distribution, an NP can be defined as follows:

> **An NP ($\mathfrak{N}$) is defined as a subset cluster within a network with ν governing transport variables where $\forall\ \upsilon \subseteq \mathfrak{N},\ \exists\ \Delta_i\ s.t.\ \sigma_i < \Delta_i, i = 1, 2, \ldots, \nu$ where σ_i is the standard deviation of the transport variable d_i for all nodes in υ and Δ_i is sufficiently small.**

In other words an NP is a patch within a network showing homogeneous transport properties. All nodes of an NP share same values for every transport dimension d_i where $i = 1, 2, \ldots, \nu$.

For this paper, we follow a powerful method presented in [10], that is able to estimate the MFPT in complex scale-invariant media, to calculate MFPT within each NP patch. Therefore, we propose a method to identify network

primitives based on the transport properties *df* and *dw* (which will be discussed in details later) to identify NPs for inhomogeneous random walks. Such an NP would therefore be a subset of network nodes where the transport properties *df* and *dw* are common at each node.

2.2. MFPT prediction in homogeneous node distributions

In this paper we use the method presented in [10] to calculate the MFPT within a given NP. We propose this method suitable for calculating MFPT between any source-target pair within an NP (where transport properties are homogeneous). In other words this method will be used to calculate MFPT in one 'hop' in the inhomogeneous network. An extension of this method which address directional bias existing in some networks is included in Supplementary Information Section S2 for completeness. We will now summarize the method from [10] for the use in hop-wise prediction.

This method makes it possible to estimate the MFPT to reach any target state starting from any source state given that the transport properties are shared for all nodes through which the random walker would traverse through. The coupled properties of the network (state space) and the random walker (system) are described using two transport properties *df* and *dw*. These two transport properties have been used in many MFPT estimation and network analysis methods found in literature [38, 10, 34, 11, 39, 40].

The *df* is a measure of the reachability of nodes for the walker and expressed in terms of the density of nodes in a network. It is commonly calculated using

$$N \propto r^{df} \tag{1}$$

where *df* gives the rate at which the number of nodes (N) increases with distance r from the source node at which *df* is calculated [10, 34, 39]. In the case of a network being continuous euclidean state space (i.e. when the random walker can take any position in a continuous state space), *df* follows the dimension of the state space. The other transport property important in estimating MFPT is a measure of the ease with which the walker can move away from the source.

This is measured in terms of dw which, for an unbiased network is calculated using

$$t_{exit} \propto r^{dw} \tag{2}$$

where t_{exit} is the exit time for a random walker to reach a sphere of radius r from the source [38, 34, 39]. In other words, dw is the rate at which the random walker moves away from the source node. For a scale-free network this is the fractal dimension of the structure formed by the random walk.

A powerful method to calculate MFPT using the relative values for df and dw for a network [34] is presented in [10] where for a length-scale-invariant network, the MFPT can be obtained using

$$MFPT \sim \begin{cases} N(A - Br^{dw-df}), & \text{if } dw < df \\ N(A + B\ln r), & \text{if } dw = df \\ N(A + Br^{dw-df}), & \text{if } dw > df \end{cases} \tag{3}$$

where N is the number of nodes, $\sim$ indicates large N asymptotic equivalence and A & B are domain dependent constants. MFPT for homogeneous anisotropic random walks can be predicted using the same equation 3 by replacing dw with a transport property called bias-modified walk dimension (dw_b) which is discussed under Supplementary Information section S2.

Given the distribution of df and dw (or dw_b) for all nodes in a network, the next question is how to identify NPs such that MFPT calculation over several NP's is possible. One main requirement is that every node in the network should belong to an NP. It is also understood that the smaller the NP's are, the higher the complexity of calculation would become. Therefore, an aim in NP identification should always be to identify NPs to be as large as possible. The next sections outline the NP identification process for a hop-wise estimation of MFPT in inhomogeneous media.

2.3. Hop-wise estimation

Getting back to the bigger picture, the final output of the calculation should be predicting the MFPT for a random walker initially at source S to reach a

known target T, irrespective of which NPs the S and T belongs to. A hop-wise calculation approach is adapted to make this possible. Figure 1 shows a network divided into 15 NPs on which a random walk takes place. Since any given random walk from S to T crosses many NP boundaries, first passage time calculation for that particular path can be broken down to n number of hops as shown in Fig. 1(a). This is possible because MFPT for any one hop can be calculated using the method described earlier. The MFPT for that particular hop sequence can be obtained from:

$$MFPT = \hat{T}_1 + \hat{T}_2 + \cdots + \hat{T}_n \tag{4}$$

where T_i is the MFPT to reach the NP exit point at the i^{th} hop. In a real prediction scenario, as there would be many possible exit points for a random walker to leave its current NP, the hop-wise prediction is done using parallel processing of many exit points weighed by a probability density distribution (P_{DD}) which is covered in detail later in the paper (Section 2.5.5 and Fig. 8). The final prediction of MFPT would be

$$< T >= \sum_{i=1}^{p} w_i MFPT_i \tag{5}$$

where p is the number of alternate paths considered and w_i is the probability of path i which is calculated from the training dataset as will be discussed later in the chapter.

Identifying NPs to be convex in shape greatly reduces the complexity of the hop-wise estimation by reducing multiple cross-over points between two adjacent NPs for any given hop, and eliminating enclosed NPs and possible infinite loops in calculation. The next section discusses the steps followed in developing an algorithm to identify NPs such that the following aims (as summarized below) are achieved.

- All nodes belonging to any given NP should be homogeneous in *df* and *dw* values (having a standard deviation below a given threshold).

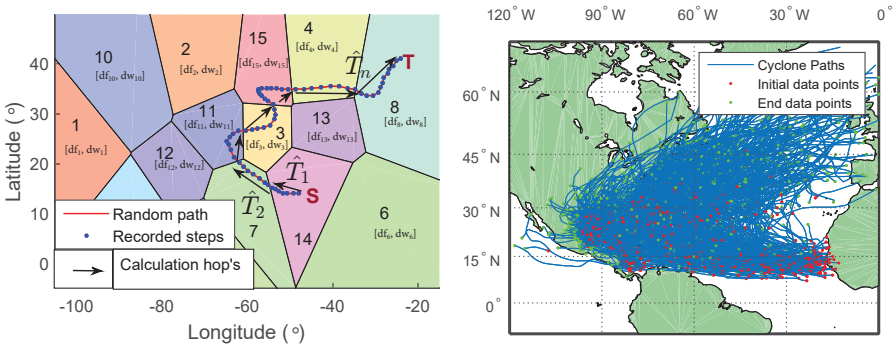

Figure 1: **Hop-wise MFPT calculation example. (a)** An example calculation of MFPT to reach a known target (T) over several NPs identified by unique $[df, dw]$ vectors. **(b)** The dataset for this example consists of the cyclone tracks over the North Atlantic ocean between 1950 and 2012. (Section 2.5.1). The division of the state-space into NPs is discussed in detail under section 2.5.3 for this particular case study.

- Every node should belong to one and only one NP (total coverage and no overlapping). $\bigvee_{i,j} NP_i \cap NP_j = \emptyset$. At this point the NPs are limited to be non-overlapping to reduce the complexity of identifying NP exit points.
- NPs should be convex (for unique exit points per hop).
- It is desirable to have larger NPs (to reduce calculation complexity).

The following sections detail the steps followed in identifying NPs taking a generic two dimensional node distribution. Nevertheless, these concepts are applicable to networks of higher dimensions as will be discussed later in the paper.

In order to describe the NP identification process described in the next section, a hypothetical 2D node distribution is introduced; hereafter called 'network Y' (Fig. 2). For simulated random walks, since df and dw calculations are sensitive to the edges of the considered portion of the network (with non absorbing boundaries), a margin around this network section is considered in the calculations (Fig. 2(a)). The transport properties for this generic discrete 2D network are calculated at each node similar to the calculation in the homogeneous hypothetical networks described earlier (detailed calculation steps

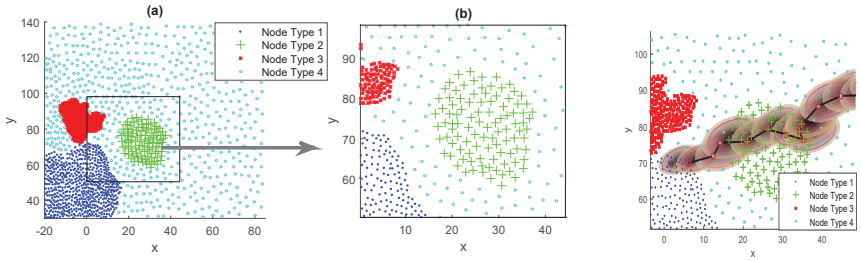

Figure 2: **Node distribution of 'Network Y'. (a)** The extended network used for getting the transport primitives. **(b)** The considered portion of Network Y. **(a)** The effect of node type on probability density distribution for next step selection. The footprint of P_{DD} differs for different node types.

are included in Supplementary information Section S1) and with a consistent and uniform bias at an angle of 45° clockwise to the north assumed. The only difference in the calculation is that as multiple node types are assumed (shown in different colors in Fig. 2), the R value (Rate of spread) of these node type differ from each other. This in turn changes the P_{DD} for selecting the next step for each node type thus making it an anomalous walk [38]. This is a common characteristic in many naturally occurring random walks and can be visualized with the example of fire propagation where depending on the fuel type (e.g. short grass, timber grass, short brush, dormant brush, hardwood litter, etc.), the rate of spread differs. Figure 2(c) shows an example random walk and P_{DD} at each step. From this it is clear how the P_{DD} changes with node type. The rest of this discussion assumes that the values of df and dw (or dw_b) are known for every node in the network.

2.4. Identifying Network Primitives (NPs)

Given the distributions of df and dw, the next step of the process is identifying the different hypotheses that exists as 'NP types' in a given network and which type each node is of. A hypothesis or NP type is a type of predefined behavior at each node and in this case is defined by a unique pair of transport variables $H_i = [df_i, dw_i]$ (Fig. 1). Depending on the complexity of

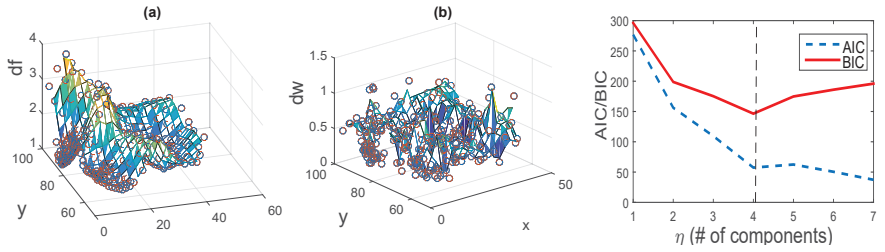

Figure 3: **Identification of NP types for network $'Y'$.**The distributions of **(a)** *df* and **(b)** *dw* with respect to the coordinate location of the node in the network. **(c)** Using AIC and BIC for selecting the number of components for GMM for NP type identification for network Y.

the network, different hypotheses can be identified either by manually observing the distributions of *df* and *dw* or by using model selection techniques such as Gaussian Mixture Models (GMM) on the same [41]. The final output will be a limited set of NP types (η) where each type is identified by a unique vector $H_i, i \in [1, 2, \ldots, \eta]$. The calculation efficiency of the proposed hop-wise MFPT prediction method is higher for networks with a smaller set of NP types (which would result in a smaller NP set).

The calculated distributions of *df* and *dw* for network Y are shown in Figs. 3(a)-(b). From this, the earlier discussion on dividing a network into homogeneous patches can be further explained. From Fig. 3(a) and 3(b) we can see that different parts of the continuous 2D network shows different transport properties. The NP identification process is how these patches can be identified.

GMM technique is often used for data clustering when clusters are of different sizes and correlation [42]. It is a parametric probability density function represented as a weighted sum of Gaussian component densities [41, 43]. GMM uses an iterative algorithm to select components to maximize of posterior probability. GMM was used for hypotheses selection for network Y where the number of hypothesis (η) was selected using both Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) to avoid any error from underesti-

mation or overestimation [44]. Figure 3(c) shows these values obtained with different number of components used in using GMM for hypothesis selection. A number of 4 NP types ($\eta = 4$) were selected considering these results and desired simplicity of real-time computation.

For network Y (η=4), the hypotheses are defined as $H_1 : H_4$. The NP type at each node was calculated using the following steps.

1. Identify the set of η number NP types/Hypotheses using GMM (i.e. identify the values of df and dw for each NP type /hypothesis).

$$\underset{i=1:\eta}{\forall} H_j = [df_j, dw_j] \tag{6}$$

2. Calculate likelihood of each node belonging to each hypothesis given the df and dw at each node. (Denoted as $P(H_i|dw)$ and $P(H_i|df); i = 1 : \eta$)

$$\begin{cases} \underset{i=1:\eta}{\forall} P(H_i|df) = \frac{\frac{1}{|df - H_i(1)|}}{\sum\limits_{i=1:\eta} \frac{1}{|df - H_i(1)|}} \\ \underset{i=1:\eta}{\forall} P(H_i|dw) = \frac{\frac{1}{|dw - H_i(2)|}}{\sum\limits_{i=1:\eta} \frac{1}{|dw - H_i(2)|}} \end{cases} \tag{7}$$

3. Get joint probability for $H_1 : H_\eta$ at each node. This is obtained by element-wise multiplication of the above likelihood matrices.

$$\underset{i=1:\eta}{\forall} \Theta_i = P(H_i|df) * P(H_i|dw) \tag{8}$$

4. Use Maximum A'Posteriori (MAP) estimate at each node to identify which hypothesis each node is most likely to be of.

$$H = \arg\max_{df,dw} \Theta \tag{9}$$

The resulting NP type distribution for network Y (i.e H)is shown in Fig. 4(a) and the next stage of the hop-wise estimation process is clustering the nodes of the same NP type into separate NPs. Following the set of aims of NPs described earlier, a heuristic approach was chosen to identify NPs where the nodes of the same NP type are initially clustered into ellipses and then optimized using self-organizing hierarchical particle swarm optimizer (HPSO) and finally broken into

convex Voronoi regions. Initial clustering of nodes (using a GUI) of same NP type into areas of elliptical shapes was done because of the convexity of the shape as well as the ease of identifying/ demarcation. Then the number, size, shape, position, and orientation of ellipses were optimized using HPSO as it is capable of handling the possible multiple local optima and converging to a global optimum through individual and social behavior of particles as well as reinitializing velocities of particles stagnated at local optima [45]. These steps are discussed in detail in the following subsections and using a case-study of cyclone tracks over the North Atlantic ocean (Section 2.5 and Figs. 6 & 7).

2.4.1. Manual demarcation

In the initial stage, the simplest approach of clustering nodes was by manually drawing ellipses over a visual graph (Fig. 4(b)) of the network in order to identify patches of nodes of similar NP types together. This gives a basis for generating the initial population to work on to optimize the NP demarcation. Ellipses are identified by (center(x,y), major axes length, eccentricity, and orientation). Figure 4(b) shows the manually selected ellipses for network Y.

2.4.2. Self-organizing Hierarchical Particle Swarm Optimization (HPSO) for optimizing elliptical NPs

Starting with the initial few roughly sketched ellipses, it was required that these clusters were optimized. For this we selected Particle Swarm Optimization (PSO) method. It is a population-based, self-adaptive search optimization technique introduced in 1995 by Kennedy and Eberhart [46]. A population of particles (defined by the dimensions) are allowed to move around the search-space until it converges to an optimal solution. With both cognitive and social components, PSO has been known to converge to a reasonably good solution very quickly [47, 45]. As there are many networks where the network connectivity pattern and the random walk process dynamics are unfolding on the same time scale [48, 49], the quick optimization process of PSO is useful when networks change in real time. HPSO was used in order to eliminate particles from

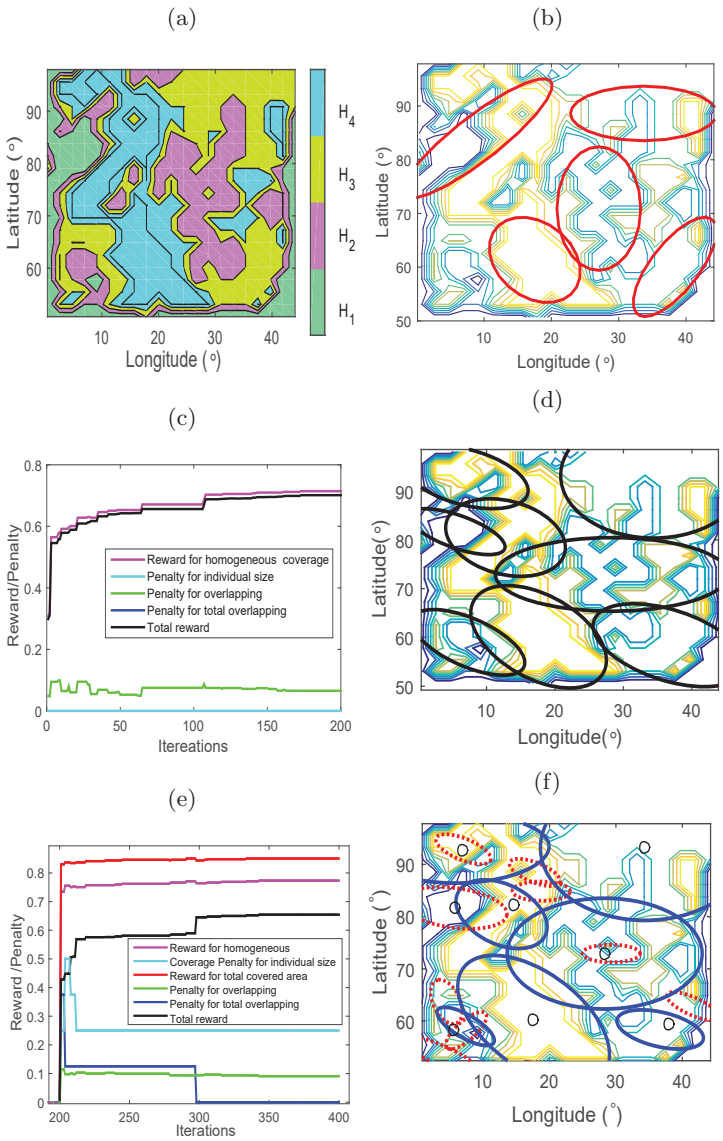

Figure 4: **NP identification steps for network Y. (a)** Distribution of NP type (hypothesis) node-wise. Every node belongs to one of node type $H_1 : H_4$. **(b)** Manual NP selection for network Y using an interactive MATLAB plot. **(c)** HPSO reward function (f_{0I}) for optimization phase I. **(d)** Identified elliptical NPs at the end of Phase I. **(e)** Reward function (f_{0II}) with respect to iterations in optimization Phase II. The sharp changes in the curves correspond to removal of smaller ellipses in subsequent iterations. **(f)** NP clusters at the end of optimization Phase II. The newly added patches are marked in dotted lines.

stagnating at suboptimal solutions [45].

In identifying NP's, HPSO was used to optimize these ellipses to maximize a reward function concerning coverage, overlap and uniformity. The optimization was done in two phases with the first phase restricting the movement of the ellipse centers and the number of ellipses. And the second phase was developed to optimize on the uniform coverage. These phases are described in detail below.

Phase I:

The centers and the ellipse numbers were kept fixed in this phase and the variables were the length of major axis, eccentricity and orientation of each of the ellipses. The total number of dimensions ($\boldsymbol{n}$) for the optimization problem is the number of variables for each ellipse (=3 for phase I) times the number of ellipses(M). The optimization problem can be stated as;

$$\begin{aligned} \min_{d_1,d_2,\ldots d_{\boldsymbol{n}}} \quad & f_0(d_1, d_2, \ldots d_{\boldsymbol{n}}) \\ \text{subject to} \quad & d_i > 0,\ i = 1, 4, \ldots, \boldsymbol{n} - 2 \qquad \text{(major axis)} \\ & 0 < d_i < 1,\ i = 2, 5, \ldots, \boldsymbol{n} - 1 \quad \text{(eccentricity)} \end{aligned} \tag{10}$$

where f_0 is the cost function and $d_1,\ d_2, \ldots,\ d_{\boldsymbol{n}}$ are the dimensions of optimization. The adaptation of HPSO algorithm to solve the optimization problem stated above is detailed in Algorithm 1. A population of $\boldsymbol{N}$ particles are initiated in $\boldsymbol{n}$ dimensions in terms of a position vector X_i and a velocity vector V_i. This initial population of particles with position vectors X_i where $i \in (1, 2, \ldots, \boldsymbol{N})$ is obtained from adding a Gaussian noise with a small standard deviation to the original particle obtained from manually drawn ellipses. In particle motion, the maximum velocity in each dimension is defined as $V_{max_d} = X_{max_d}$ as usually done in PSO to reduce roaming outside the search-space [45]. The initial velocities are defined as fractions of V_{max}. The best localization of each particle P_{id} (which shows the highest fitness), and localization of fittest particles so far (P_g) are updated with each iteration. The new positions of each particle are updated with new velocities as shown in Algorithm 1.

Algorithm 1 (HPSO in Phase I)

begin

Initialize population of $\boldsymbol{N}$ *particles*

$X_1 = [X_{maj1}, X_{ecc1}, X_{ori1}, ...X_{majM}, X_{eccM}, X_{oriM}]$ ▷ Position vector for 1st particle

$\underset{2<j<\boldsymbol{N}}{\forall} X_j = X_1 + [\epsilon_{1j}, \epsilon_{2j}, \ldots, \epsilon_{3Mj}], \epsilon \sim \boldsymbol{N}(0, \sigma^2)$ ▷ Add noise to get other particles

$V_{max} = max(X_j)$ ▷ Initialize velocity matrix

$\underset{j}{\forall} V_j = 0.1\ V_{max}$

Define

fitness threshold = f_{th}

$f_{0I} = 0, F = [0\ 0\ \ldots\ 0]_{1\times\boldsymbol{N}}$

maximum iterations = I_{max}

$[c1_{min}, c1_{max}, c2_{min}, c2_{max}] = [0.5, 2.5, 0.5, 2.5]$ ▷ Acceleration coefficients

$[w_1, w_2, w_3, w_4]$ ▷ Reward function weights

while f< f_{th} & $I < I_{max}$ **do**

$c1 = (c1_{min} - c1_{max})I/I_{max} + c1_{max}$;

$c2 = (c2_{max} - c2_{min})I/I_{max} + c2_{min}$;

for $i = 1 : \boldsymbol{N}$ **do**

$f_{0I}(x) = w_1 R_{UC} - w_2 P_{IS} - w_3 P_O - w_4 P_{TO}$ ▷ Calculate fitness (Eq.11)

Update P_{id} ***and*** P_g

if $F_i < f_{0I}(x)$ **then**

$F_i = f_{0I}(x)$ ▷ Update best fitness matrix

$P_{id} = X_i$ ▷ Best position of each particle

if $indexOF(max(F)) = i$ **then** ▷ Fittest particle so far

$P_g = P_{0I}$

end if

end if

for $d = 1$: 3M **do**

Update velocity

$V_{id} = c_1 \times \text{rand}(1) \times (P_{id} - X_{id}) - c_2 \times \text{rand}(1) \times (P_{gd} - X_{id})$ ▷ New velocity

if $V_{id} = 0$ **then**

if rand(1) < 0.5 **then**

V_{id} = rand(1) x $V_{max}(d)$

else

V_{id} = (-)rand(1) x $V_{max}(d)$

end if

end if

V_{id}=sign(V_{id}) $\times$ min(abs(V_{id}),V_{max})

$X_{id} = X_{id} + V_{id}$

if limits violated **then**

reset X_{id}

end if

end for

end for

end while

Reward function for optimization phase I (f_{0I})

The objective of phase I was to identify the main clusters which would form the largest NPs. This is an important step as the final MFPT estimation would greatly depend on the accuracy of calculation within these largest NPs. With this objective in mind, the rewards function for this phase (f_{0I}) was formulated as a weighted average for the following.

- Reward for uniform coverage (i.e the proportion of nodes covered by a single ellipse) ; R_{UC}
- Penalty for the number of overlapped nodes ; P_O
- Higher Penalty for ellipses that are totally overlapped by others; P_{TO}
- Penalty for individual size of ellipses being below a threshold ; P_{IS}

The fitness for each of the $\boldsymbol{N}$ particles are calculated for phase I as

$$f_{0I}(x) = w_1 R_{UC} - w_2 P_{IS} - w_3 P_O - w_4 P_{TO} \tag{11}$$

where $f_{0I}(x_i)$ is the fitness of the i^{th} particle (x_i) , $w_1 : w_4$ were the weights for each reward or penalty. The weights $w_1 : w_4$ of the rewards and penalties used were selected by observing the convergence rate and fitness for a range of possible fitness functions. The reward function was used in our simulation with weights as $[w1, w2, w3, w4] = [1, 0.4, 0.2, 0.4]$. Figure 4(c) shows the individual rewards and the total reward function with respect to the iteration numbers and Fig. 4(d) shows the NP cluster ellipses at the end of Phase I.

Phase II : HPSO for optimizing coverage and identifying smaller primitives

At the end of first optimization phase the main (largest) NP's are identified roughly. The next stage is concerned with clustering the isolated nodes outside of the first set of ellipses either by merging into larger NPs or by forming smaller NP's by themselves. A function called 'grains' that is used in MATLAB for image processing for clustering similar areas to identify object boarders was used to identify the centroids of uncovered areas. The next generation of ellipses were initialized partially with these centroids and the rest with the output of the earlier phase. A modified reward function was used in this stage to give more prominence to total coverage and added penalties for uncovered area. The reward function for this phase was

$$f_{oII}(x) = w_1 R_{UC} + w_2 R_{IS} - w_3 P_O - w_4 P_{IS} - w_5 P_{TO} \tag{12}$$

where R_{IS} was the reward for individual NP sizes. The weights of the individual rewards were $([w_1, w_2, w_3, w_4, w_5] = [0.33, 0.67, 0.5, 0.5, 0.5])$. Sparsely interlaced iterations were used to identify and delete ellipses smaller than a fixed lower limit. Iteration outputs for optimization phase II for network Y shown in Fig. 4(f) and Fig. 4(e) gives the individual and total reward functions (f_{oII}) with respect to iterations.

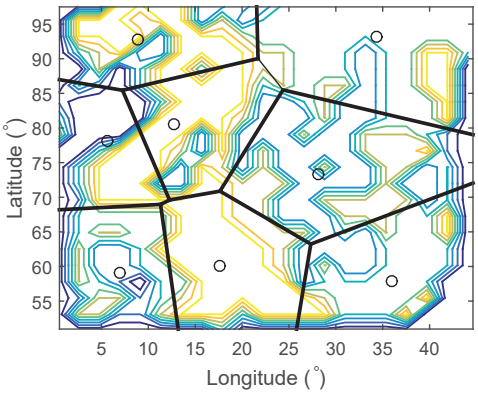

Figure 5: Final demarcation of identified NPs for network Y through Voronoi regions with the infinite edges clipped off with a network boarder. The contour plot shows the NP type distribution.

Final demarcation of NPs using Voronoi regions

The ellipse set resulting from HPSO consists of ellipses of considerable size and ones which, if considered as individual NPs are very small and will increase the MFPT estimation complexity largely. Therefore to be practical in real-time computation, the resultant set of elliptical NPs are divided into two groups depending on their coverage areas. The threshold is taken as having a major axis at least three times the rate of spread. This was done in the sense that if the walker is able to reach the furthermost end of the primitive in less than three steps, such NPs are considered inefficient by increasing calculation complexity. Nodes within ellipses such as these were merged into the NPs closest to them. The centers of gravity for larger ellipses were adjusted by a weighted average with the positions of added clusters.

Finally the total continuous area was divided into convex NPs using the centers of gravity for the remaining ellipses as centers of Voronoi regions. The last step was to clip off the infinite edges of Voronoi regions by a network boarder in order to make hop-wise MFPT estimation feasible. The final NP demarcation for network Y is depicted in Fig. 5 with a total of eight NPs identified.

This method of NP identification is presented for a 2D unconnected node distribution but the concepts are adaptable for higher dimension and even complex connected networks.

Transport properties within an NP

Once the network is divided into non-overlapping NP's, common transport properties (i.e. df and dw values) for each NP are obtained. Calculating the MFPT for a random walker to reach any target T within the same NP as the source point (i.e one hop) is straight forward using methods introduced in Section 2.2 given that the values for df and dw and constants A and B in equation (3) are known. If a different method is used for computing the MFPT, this calculation is still straight forward if the transport properties for that method were used in identifying NPs.

Ideally every node within an NP would have common values for df, dw and bias (θ) and thereby A and B. Since the NP identification process needs to allow some robustness when dealing with real world networks, there could be slight variations (below a predefined threshold) in these values between nodes in the same NP. Yet, it is important for the hop-wise MFPT calculation that common values for these dimensions are found for each NP such that equation (3) can be applied. One way of obtaining these values are using the df and dw values set by the hypothesis of the most common NP type of within the NP. But to add robustness through customization, the transport variables $df, dw, \theta, A(\theta), B(\theta)$ for each of the primitives were recalculated dividing the dataset between the NPs and again into Ψ number of angle segments since we assume directional bias in this example i.e. $\theta \in \Phi \mid \Phi\{(-\pi, -\pi + 2\pi/\Psi), (-\pi + 2\pi/\Psi, -\pi + 4\pi/\Psi), ...(-\pi + 46\pi/\Psi, -\pi + 48\pi/\Psi)\}$. It was decided to use $\Psi = 24$ angle segments for the simulations after compromising between (limited) training dataset size and accuracy. Using angle segments is only required for random walks showing directional bias..

In the next section the NP identification process is used and extended to the

hop-wise MFPT prediction with the case study of cyclone motion in the North Atlantic Ocean.

2.5. Case study : MFPT predictions for cyclone motion in the North Atlantic ocean

The methods developed to estimate MFPT for a random walker at a known source point S to reach a given target point T in inhomogeneous network through identifying NPs are now confirmed using a case study of predicting MFPT for cyclone motion using a dataset of the paths followed by cyclones over the North Atlantic ocean. This particular dataset was selected due to the fact that with respect to cyclone track history over other oceans, the dataset over NA Ocean shows a clear correlation with the Beta effect thus becomes a good candidate for an example biased random walk on inhomogeneous media.

2.5.1. Dataset

To train and validate the proposed model, we used the complete dataset of cyclone track archive for the cyclones observed over the North Atlantic ocean since 1950 from HURDAT2 ([50]) dataset. HURDAT2 is a publicly available dataset providing a record of tropical cyclones (TCs) in the North Atlantic Ocean observed from 1851 to 2013, available on-line at `http://www.nhc.noaa.gov/data/#hurda`. We used the recorded data of all cyclones above the NA ocean from 1950 to 2011 as the training dataset for our simulations. The dataset consists of 6 hourly coordinate locations of the eye of the cyclone for every documented cyclone in the categories of Tropical Depression, Tropical Storm, Hurricane, Extra tropical cyclone, Subtropical cyclone, Low, Tropical wave and Disturbance. Although the complexity of the atmosphere and stochastic nature of the cyclone motion itself creates an uncorrelated set of random walks, the biasing factor imposes broad limitations to the movement. The selected dataset is shown in Fig. 1(b) and Fig. 6 discusses how the dataset consists of a network inhomogeneous in transport properties. It is seen how the bias direction (directional inhomogeneity) and bias intensity(inhomogeneity of density of nodes) is

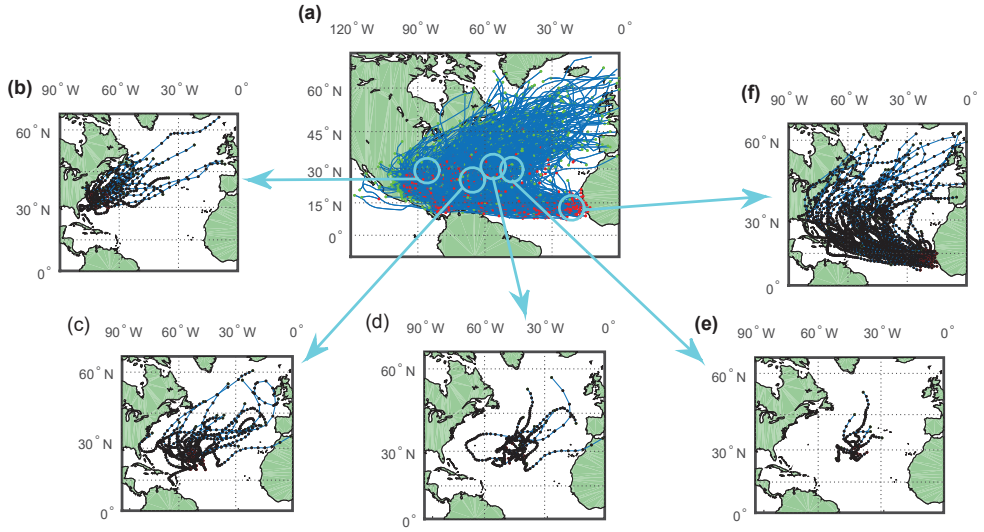

Figure 6: **The dataset of past cyclone tracks used for the case study, showing inhomogeneous transport characteristics. (a)** the complete dataset of archived cyclone track data over the NA ocean since 1950. **(b)-(f)** Subsets of tracks filtered by the initial track location. The dots show 6 hourly positions of the eye of the cyclones. It is seen how the bias direction (directional inhomogeneity) and bias intensity(inhomogeneity of density of nodes) is changing with location thus dividing the network into different NPs.

changing with location.

2.5.2. Transport Properties for case study

The state space of the selected set of random walkers is a continuous two-dimensional Euclidean space (i.e. the eye of the cyclone can take any position within the continuous state space) and therefore *df* is obtained straight forward as equal to 2 from the dimensions of the state space. Using the dataset of cyclones between 1950-2012 as the training dataset, *dw* was calculated using equation (2) following Algorithm S1 in the Supplementary Information. When calculating *dw* at different points in the state space, it was observed that the values vary considerably throughout the considered area (i.e. state space). In order to get the distribution of *dw*, the area was divided into a grid-world and used the datasets of cyclones originating from each grid cell to calculate *dw* with

respect to direction.

2.5.3. NP identification

Similar to network Y, the dw and df profiles were used to get the NP type / hypothesis at each location. Due to the network being continuous, it was required to use the above mentioned grid-world to NP type identification. Also due to the state space being continuous, the df remained constant for every point in the state space and therefore it was only required to identify hypotheses for dw, as it was the only variable transport property of importance for the calculation. Similar to network Y, the dw profile used for NP type selection was composed of dw in the direction of maximum bias (Supplementary information section S2).

As the NP Types were only dependent on dw for this example, hypothesis identification was done manually by observing the dw distribution. Four hypotheses were identified from this distribution which fixed the NP type set at $\eta = 4$. The point-wise NP type distribution is shown in Fig. 7(b). NP identification process for this case study follows the procedure explained with the example of network Y and the outputs and intermediate steps are illustrated in Fig. 7.

A total number of 15 convex NPs were obtained using the 4 identified hypotheses. The values for df and dw for each of the 15 NPs were identified by using the dataset divided by the NP in which each cyclone was originally recorded in. (i.e. from the initial recorded position). A unique value set of $\{df, dw(\alpha), A(\alpha), B(\alpha), \alpha\}$ where α is the direction of the hop is stored for each NP for the real-time MFPT prediction for each hop using equation (3). The angles were divided into $\Psi = 24$ segments limited by training dataset. These values were stored in a cell structure to be accessed in real-time computation.

2.5.4. Hop-wise MFPT estimation

Once the NPs are identified with their inherent (to the network and the walker) transport properties, a hop-wise MFPT prediction method was devel-

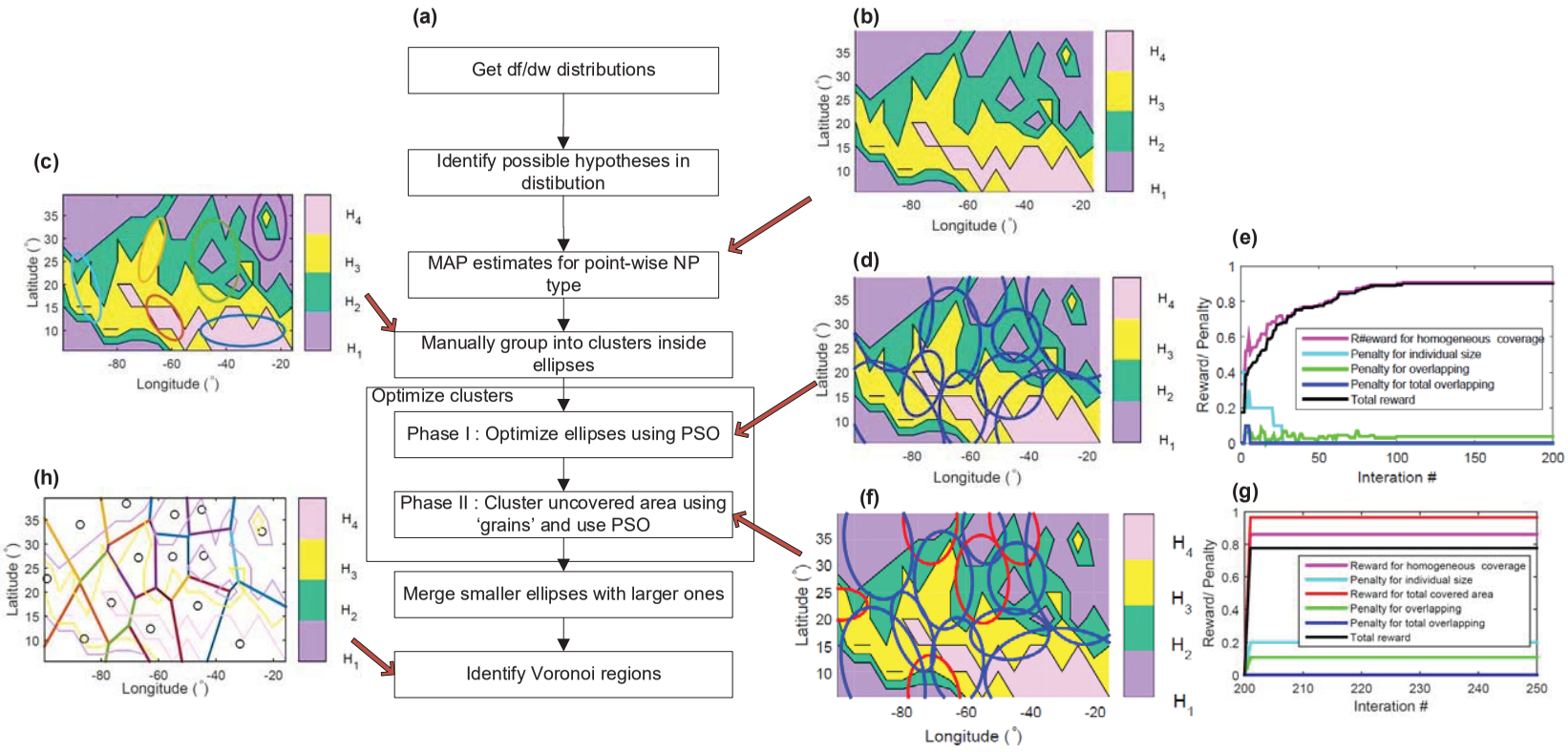

Figure 7: **Identifying NPs for the case study of cyclone tracks over the NA ocean.** **(a)** The flowchart of NP identification process. **(b)** Point-wise NP type/hypothesis distribution **(c)** Manual demarcation of elliptical NP patches. **(d)-(e)** Resulting NP distribution at the end of optimization phase I, and its reward function. **(f)-(g)** The same for phase II. **(h)** Identified NP set.

oped. Unlike the point to point estimation as in the case of a homogeneous network where there was only one pair of S and T considered by the walker in a single attempt, when the network is divided into patches with different characteristics, the paths a walker would take in the next 'hop' differ largely depending on the exit point of the previous NP. Therefore, it was required that multiple exit points were selected for each starting points within the current NP. This as a result will increase the calculations exponentially with each 'hop'. The scalability of the proposed methods will be discussed later using calculation complexities.

The first step in the hop-wise estimation is selecting the exit points for each NP the walker enters (given that the final target point T, is not inside the NP the walker enters). There is one assumption in this hop-wise MFPT calculation that if the random walks are directionally biased, there are no loops in the bias. If that was the case, the calculation would crash in an infinite loop until the maximum hops are reached. Although it is possible to extend the calculation to include this case, due to the limitation of computational resources, this assumption is enforced as a limitation to the number of hops. Alternatively, the limit could be enforced as a limitation to the variation of MFPT with additional iterations.

In getting the actual exit points for a random walker to move out of its current NP, it is straight forward for a discrete network by selecting the set of nodes closest to the boundary of the NP. When the network is continuous, it was required to find a discrete number of exit points from the current NP. This number is a compromise between the required precision and the affordable computational complexity. In the case study with directionally biased random walks, in order maximize the utilization of knowledge encapsulated in transport variables calculated from previous data, an exit point was used for each angle segment transport variables are pre-calculated for (Φ) and which are within the probabilistic region. In order to get the possible exit angle segment imposed by the network bias, this bias direction is the most common exit direction from the NP. If the bias intensity is also known, the size of the angle segment of

possible exit points can also be calculated. Without a loss of generality, an angle segment of ϕ was chosen around the bias direction with 95% confidence interval that random walkers in that NP would exit through. This was obtained from the training dataset used in getting the transport variables. This process is not necessary for isotropic random walks.

2.5.5. Probabilistic path selection

In the existence of directional bias, whichever method was used to get the set of transit/exit points for the random walker from its current NP, it is inherent in its biased motion that it is more likely that some paths are taken than others. The probability of exiting through a node directly in the direction of bias is more likely than in other directions. Therefore, probability of path selection obtained from past data, is integrated into the hop-wise MFPT prediction.

For any random walk, the probability of any given random path being the actual random walk is the joint probability along the hops of that path. Thus, the final MFPT estimation is an average MFPT weighted by the probability of that particular path.

Figure 8 shows a few examples of the selected hop 'branches' for a selected source and target set for up to three hops. It is worth noting that the plots are not random walks but rather the connectivity of a few calculation hops of random walks. Referring back, Fig. 1 showed one such path.

2.5.6. Parallel processing in Hop-wise MFPT prediction

As described earlier, one drawback in the hop-wise MFPT prediction is that it needs to consider many alternate paths of a single random walk depending on transit points between NPs. This increases the computations required exponentially with each NP interface a random walker needs to cross in order to reach the target point. This makes computational resources a bottleneck for the estimation. But an important factor in this calculation is that although multiple paths of a single random walker has to be considered, these multiple paths are independent of each other because in reality although multiple paths

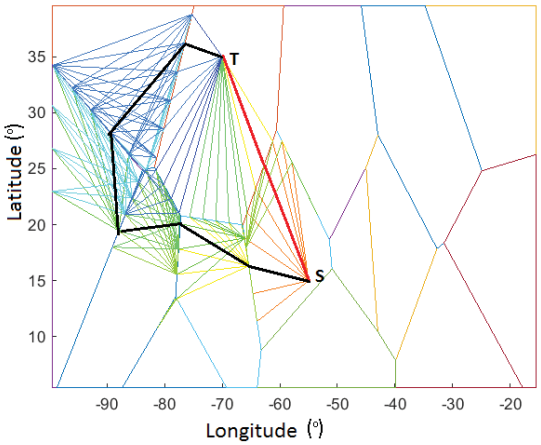

Figure 8: Hop-wise MFPT estimation for case study of cyclone tracks over NA Ocean. The graphs show a few 'branches' of hop-wise routes considered in the MFPT estimation process. Different hops are shown in different colors and two sample routes (collection of hops) are highlighted in red and black.

are considered a walker can only take one at a given time. To make use of this independence, the MATLAB code developed for this calculation uses parallel processing functions to improve calculation efficiency. To preserve the efficiency provided by parallel processing, variables used in each parallel process needs to be independent from each other. Therefore parallel processing was used only per hop to calculate MFPT via multiple exit/transit nodes which are independent.

The real time computational complexity for the hop-wise prediction is $\mathcal{O}(n^h)$ where n is the total number of dimensions of each particle and h is the maximum number of hops considered. Therefore with scaling in terms of higher dimension random walks, the real time computational complexity will not be effected. However, in obtaining *df* and *dw* distributions, the calculation complexity will exponentially increase with dimensions of the state space. Yet, the computational time could be significantly reduced by parallel processing as calculations are still mutually independent.

2.5.7. Final MFPT estimation

The MFPT estimates obtained for each hop of the calculation were stored in a MATLAB cell structure until the final estimation converges or if the hop number has reached a threshold. When the networks are biased and when the bias does not form loops, after a limited number of steps, if the target is not reached, it can be assumed that the MFPT tends to infinity. This number of steps is reduced with the increase in bias intensity. With the number of iterations set to the converging threshold of the network, Algorithm 2 presents the pseudo code followed in the final MFPT estimation process.

Algorithm 2 Hop-wise MFPT calculation

Begin

Define $S, T, frontline = S$ ▷ Input Source and Target

Load $NPs, df, dw, A, B, P_{DD}$ ▷ Import NP & Transport properties (calculated off-line)

Identify NP containing $S\&T \rightarrow NP_S, NP_T$

set h_{max} ▷ maximum number of hops

if $S == T$ **then**

$hops\{1\} = \{source = S, Pr = 1, mfpt = 0, prev_{NP} = 0, reached = 1\}$

else

$hops\{1\} = \{source = S, Pr = 1, mfpt = 0, prev_{NP} = 0, reached = 0\}$

for $hop = 2 : h_{max}$ **do**

for n=1:length(frontline) **do** ▷ For every boundary node from previous hop

if $hops\{hop\}.reached > 0$ **then**

Append $hops[hop]|hops[hop-1](n)$ ▷ Append n^{th} elements

else

$[newfrontline, mfpt_{all}, NP_S, reached_{all}] = getNexthop$

$Pr_1 = P_{DD}\{NP_S\}.hops\{hop-1\}.Pr(n)$ ▷ Probability per path

Append $hops\{hop\}|\{source = frontline, Pr = Pr_1, mfpt = mfpt_{all}, reached = reached_{all}, prev_{NP} = NP_S\}$

```
            end if
        end for
        frontline = newfrontline;
    end for
end if
```

---

getNexthop

```
if NP_S == NP_T then
    calculate final MFPT using equation 3)
    return T, mfpt, NP_S, reached = 1
else
    Identify exit points from NP_S → frontline
    Start ‖ processing
    for every point on the previous frontline do
        if target reached then
            calculate final MFPT using P_r
            return T, mfpt, NP_S, reached = 1
        else
            calculate new frontline
            calculate MFPT to reach every i^th point on frontline
            return {frontline, mfpt, NP_S, reached = 0}
        end if
    end for
    End ‖ processing
end if
```

2.5.8. Hop-wise MFPT estimation

For the cyclone track dataset, by observation the number of maximum hops was limited to $h_{max} = 20$ as the results generally converged by 10 hops. Figure 9 shows the MFPT prediction for ten (S,T) pairs with respect to the number of hops considered (here the branches which have not yet reached the target are

considered as misses). The (S, T) pairs are the same as the ten cases presented in Table 3. It can be concluded that after 20 hops, any walkers which have not yet reached the target have indeed missed it as the probability of returning against the bias path is extremely low (given that there are no loops in the bias vector field as in this example). If there were loops in the bias vector field, convergence would be delayed.

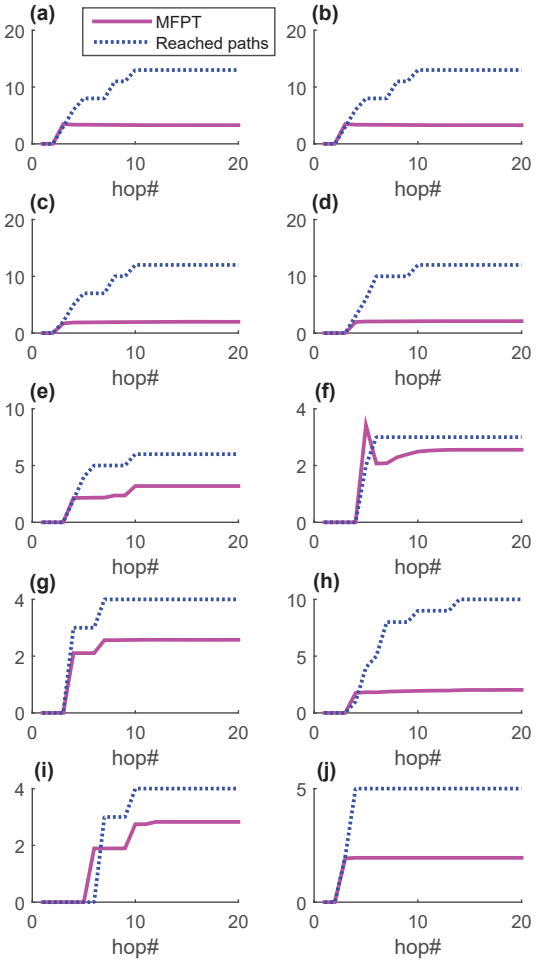

Figure 9: MFPT estimation (in $\#days$) for the 10 presented comparisons in Table 3 for the cyclone dataset. The convergence of the iterative MFPT estimation is observed.

3. Results and Discussion

Comparisons were made for hop-wise calculations to real data and predictions assuming homogeneous transport properties for many (S,T) pairs using methods from [10].Comparisons were done with only larger distances between S and T where the hop-wise calculation made sense (for S and T to be in different NPs). Currently 120 h is the maximum lead time official forecasts can provide for cyclone tracks. Ten comparisons are presented in Table 3 and from this it is shown that breaking the problem into NPs reduces the average error from 18.8% to 5.4% with respect to the homogeneous network assumption, but at the cost of 12 times increase in processing time required. However, this increase in processing time is subjective and the importance changes with the application. For example, in the cyclone track dataset, new data points were available every 6 hours and the increase in processing time was in the order of one or two minuets does not make any difference. It was also noted that the processing time was dependent on the resolutions used. Figure 9 presents the MFPT convergence rate with the number of calculation 'hops'. It can be observed that with the increase of the number of hops required to reach the target, the effect the path has on the final MFPT reduces. All simulations and calculations in this paper were done using MATLAB coding on an Intel core^{TM} i7-4770 CPU with 16 GB available RAM. In conclusion, while requiring more computational power, breaking the problem of MFPT estimation on an inhomogeneous network by dividing the network using the concept of Network Primitives increases the estimation accuracy considerably.

4. Conclusions

This paper investigated the possibility of estimating MFPT between any given source target pair for a random walker in media which is inhomogeneous in transport characteristics. A hop-wise calculation method was adapted by partitioning the network into NPs homogeneous in network properties (*df* and

Test case	**Source [Lat,Lon]**	**Target [Lat,Lon]**	**Real Data (days)**	**Hop-wise Prediction**		**Prediction assuming network homogeneity**	
				Time (days)	$\|Error\|$ **%**	**Time (days)**	$\|Error\|$ **%**
1	[12 -20]	[32 -70]	11.2917	11.3066	0.13%	10.6305	5.86%
2	[15 -55]	[35 -70]	7.25	7.8085	7.70%	10.2541	41.44%
3	[15 -60]	[32 -70]	6.0417	6.0417	0.00%	7.712	27.65%
4	[15 -58]	[23 -64]	7.4722	7.9218	6.02%	5.3626	28.23%
5	[15 -50]	[25 -60]	8.3611	8.3791	0.22%	6.8037	18.63%
6	[18 -45]	[27 -67]	8.0278	8.3881	4.49%	9.4525	17.75%
7	[12 -45]	[35 -65]	11.8333	12.7713	7.93%	11.0873	6.30%
8	[13 -21]	[29 -69]	10.8333	10.2158	5.70%	10.4282	3.74%
9	[14 -48]	[34 -69]	9.7222	10.3003	5.95%	10.6472	9.51%
10	[20 -45]	[25 -65]	9.6111	8.119	15.52%	6.829	28.95%

Table 3: **Comparisons of MFPT predictions with actual average FPTs for the dataset of past cyclone tracks over the NA ocean discussed in section 2.5.1.** A set of 10 source and target sets are presented to compare the average FPTs with the hop-wise prediction proposed in this paper, and with the predictions obtained using methods discussed in Section 2.2 assuming homogeneous transport properties. It is shown that breaking the problem into NPs using the proposed methods reduces the average error from 18.8% to 5.4% with respect to the homogeneous network assumption.

dw). An algorithm was developed to identify convex NPs for a 2D node distributions which was optimized using HPSO. This can be extended to higher dimensions with the use of higher dimension ellipsoids and Voronoi regions. Parallel processing computation methods were adapted to improve on the real-time calculation efficiency. The methods are described initially with simulated random walks on a hypothetical 2D discrete node distribution and then using the dataset of past cyclone tracks over the North Atlantic Ocean. The methods were verified by comparing results with real data and estimations assuming homogeneous transport properties. Comparison results showed a 3.5 times less error with the proposed hop-wise approach. One limitation with the method is that when there is zero prior knowledge of random walks in an NP in some direction, this method cannot be used to predict for that direction (as no transport variables can be calculated). However, for the example of predicting MFPT for cyclone motion, future work could include using simulated data by physical modeling of the complex processes governing the large-scale and inner-core characteristics of TCs and their interaction with the environment to derive the transport variables. Finally, it is worth noting that this method of using multiple possible exit points from an NP can be directly used to extend this method in calculating MFPT for random spreads (i.e. branching random walks) such as fire propagation.

[1] S. Redner, A guide to first-passage processes, Cambridge Univ Pr, 2001.

[2] L. Lovsz, Random walks on graphs, Combinatorics, Paul erdos is eighty 2 (1993) 1–46.

[3] R. A. Meese, K. Rogoff, Empirical exchange rate models of the seventies: Do they fit out of sample?, Journal of international economics 14 (1) (1983) 3–24, meese1983.

[4] J. S. Gall, I. Ginis, S.-J. Lin, T. P. Marchok, J.-H. Chen, Experimental tropical cyclone prediction using the gfdl 25-km-resolution global atmospheric model, Weather and Forecasting 26 (6) (2011) 1008–1019.

[5] I. Wijesundera, M. N. Halgamuge, T. Nirmalathas, T. Nanayakkara, A computationally efficient framework for stochastic prediction of flood propagation, in: Information and Automation for Sustainability (ICIAfS), 2012 IEEE 6th International Conference on, IEEE, pp. 25–28.

[6] V. Isham, S. Harden, M. Nekovee, Stochastic epidemics and rumours on finite random networks, Physica A: Statistical Mechanics and its Applications 389 (3) (2010) 561–576.

[7] S. Karlin, A first course in stochastic processes, New York, Academic Press c1969., 1966.

[8] S. Karlin, H. E. Taylor, A second course in stochastic processes, Elsevier, 1981.

[9] E. Agliari, R. Burioni, Random walks on deterministic scale-free networks: Exact results, Physical Review E 80 (3) (2009) 031125.

[10] S. Condamin, O. Benichou, V. Tejedor, R. Voituriez, J. Klafter, First-passage times in complex scale-invariant media, Nature 450 (7166) (2007) 77–80.

[11] C. Song, S. Havlin, H. A. Makse, Origins of fractality in the growth of complex networks, Nature Physics 2 (4) (2006) 275–281.

[12] B. Meyer, E. Agliari, O. Bnichou, R. Voituriez, Exact calculations of first-passage quantities on recursive networks, Physical Review E 85 (2) (2012) 026113.

[13] B. Kahng, S. Redner, Scaling of the first-passage time and the survival probability on exact and quasi-exact self-similar structures, Journal of Physics A: Mathematical and General 22 (7) (1989) 887.

[14] C. Song, S. Havlin, H. Makse, Self-similarity of complex networks, Nature 433 (7024) (2005) 392–395.

[15] C.-L. Pu, W.-J. Pei, Mixing search on complex networks, Physica A: Statistical Mechanics and its Applications 389 (3) (2010) 587–594. doi:http://dx.doi.org/10.1016/j.physa.2009.10.007.
URL http://www.sciencedirect.com/science/article/pii/S0378437109008565

[16] T. Koren, A. Chechkin, J. Klafter, On the first passage time and leapover properties of lvy motions, Physica A: Statistical Mechanics and its Applications 379 (1) (2007) 10–22.

[17] V. Colizza, A. Vespignani, Invasion threshold in heterogeneous metapopulation networks, Physical Review Letters 99 (14). doi:ARTN148701DOI10.1103/PhysRevLett.99.148701.
URL <GotoISI>://WOS:000249974000085http://journals.aps.org/prl/abstract/10.1103/PhysRevLett.99.148701

[18] J. P. Aparicio, M. Pascual, Building epidemiological models from r0: an implicit treatment of transmission in networks, Proc Biol Sci 274 (1609) (2007) 505–12.
URL http://www.ncbi.nlm.nih.gov/pubmed/17476770

[19] L. Giuggioli, S. Prez-Becker, D. P. Sanders, Encounter times in overlapping domains: application to epidemic spread in a population of territorial animals, Physical Review Letters 110 (5) (2013) 058103–058103.

[20] L. Gallos, C. Song, S. Havlin, H. Makse, Scaling theory of transport in complex biological networks, Proceedings of the National Academy of Sciences 104 (19) (2007) 7746.

[21] J. K. Rudra, J. J. Kozak, Spectral dimension of regular and fractal lattices, Physics Letters A 151 (8) (1990) 429–432.

[22] S. Hwang, D. S. Lee, B. Kahng, First passage time for random walks in heterogeneous networks, Physical review letters 109 (8) (2012) 088701, pRL.

[23] T. Nanayakkara, K. Byl, L. Hongbin, S. Xiaojing, T. Villabona, Dominant sources of variability in passive walking, 2012 IEEE International Conference on Robotics & Automation (2012) 100386559908 2012 IEEE International Conference on Robotics & Automation.

[24] C. Pu, S. Li, A. Michaelson, J. Yang, Iterative path attacks on networks, Physics Letters A 379 (28) (2015) 1633–1638.

[25] C. Pu, S. Li, J. Yang, Epidemic spreading driven by biased random walks, Physica A: Statistical Mechanics and its Applications 432 (2015) 230–239.

[26] C.-L. Pu, S.-Y. Zhou, K. Wang, Y.-F. Zhang, W.-J. Pei, Efficient and robust routing on scale-free networks, Physica A: Statistical Mechanics and its Applications 391 (3) (2012) 866–871.

[27] C.-L. Pu, J. Yang, W.-J. Pei, Y.-T. Tao, S.-H. Lan, Robustness analysis of static routing on networks, Physica A: Statistical Mechanics and its Applications 392 (15) (2013) 3293–3300.

[28] V. Kurella, J. C. Tzou, D. Coombs, M. J. Ward, Asymptotic analysis of first passage time problems inspired by ecology, Bulletin of mathematical biology 77 (1) (2015) 83–125.

[29] A. Godec, R. Metzler, Optimization and universality of brownian search in quenched heterogeneous media, arXiv preprint arXiv:1503.00558.

[30] J. D. Noh, H. Rieger, Random walks on complex networks, Physical review letters 92 (11) (2004) 118701.

[31] V. Sood, S. Redner, D. Ben-Avraham, First-passage properties of the erd ”osrenyi random graph, Journal of Physics A: Mathematical and General 38 (1) (2005) 109.

[32] C.-L. Pu, W.-J. Pei, A. Michaelson, Robustness analysis of network controllability, Physica A: Statistical Mechanics and its Applications 391 (18) (2012) 4420–4425.

[33] V. Tejedor, O. Bnichou, R. Voituriez, Close or connected: Distance and connectivity effects on transport in networks, Physical Review E 83 (6) (2011) 066102.

[34] M. F. Shlesinger, Mathematical physics: First encounters, Nature 450 (7166) (2007) 40–41.

[35] I. Wijesundera, M. N. Halgamuge, T. Nirmalathas, T. Nanayakkara, A geographic primitive-based bayesian framework to predict cyclone-induced flooding*, Journal of Hydrometeorology 14 (2) (2013) 505–523.

[36] I. Wijesundera, Estimation of mean first passage time (mfpt) in naturally biased inhomogeneous environments, Thesis (2015).

[37] N. M. Ferguson, M. J. Keeling, W. J. Edmunds, R. Gant, B. T. Grenfell, R. M. Amderson, S. Leach, Planning for smallpox outbreaks, Nature 425 (6959) (2003) 681–685. doi:DOI10.1038/nature02007.
URL <GotoISI>://WOS:000185924500032

[38] D. Ben-Avraham, S. Havlin, Diffusion and reactions in fractals and disordered systems, Cambridge Univ Pr, 2000.

[39] V. Tejedor, Random walks and first-passage properties, Thesis (2012).

[40] C. Meja-Monasterio, G. Oshanin, G. Schehr, First passages for a search by a swarm of independent random searchers, Journal of Statistical Mechanics: Theory and Experiment 2011 (06) (2011) P06022.

[41] D. Reynolds, Gaussian Mixture Models, Springer US, 2009, book section 196, pp. 659–663. doi:10.1007/978-0-387-73003-5_196.

[42] Matlab r2015b documentation.
URL http://au.mathworks.com/help/stats/clustering-using-gaussian-mixture-models.html?s_tid=gn_loc_drop

[43] M. Gonzlez-Fierro, D. Hernndez-Garca, T. Nanayakkara, C. Balaguer, Behavior sequencing based on demonstrations: a case of a humanoid opening a door while walking, Advanced Robotics 29 (5) (2015) 315–329.

[44] G. Celeux, G. Soromenho, An entropy criterion for assessing the number of clusters in a mixture model, Journal of classification 13 (2) (1996) 195–212.

[45] A. Ratnaweera, S. K. Halgamuge, H. C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, Evolutionary Computation, IEEE Transactions on 8 (3) (2004) 240–255.

[46] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Neural Networks, 1995. Proceedings., IEEE International Conference on, Vol. 4, pp. 1942–1948 vol.4. doi:10.1109/ICNN.1995.488968.

[47] R. C. Eberhart, Y. Shi, Particle swarm optimization: developments, applications and resources, in: Evolutionary Computation, 2001. Proceedings of the 2001 Congress on, Vol. 1, IEEE, pp. 81–86.

[48] N. Perra, A. Baronchelli, D. Mocanu, B. Gonalves, R. Pastor-Satorras, A. Vespignani, Random walks and search in time-varying networks, Physical review letters 109 (23) (2012) 238701.

[49] Y. Sun, M. Dai, L. Xi, Scaling of average weighted shortest path and average receiving time on weighted hierarchical networks, Physica A: Statistical Mechanics and its Applications 407 (2014) 110–118. doi:http://dx.doi.org/10.1016/j.physa.2014.03.088.
URL http://www.sciencedirect.com/science/article/pii/S0378437114002994

[50] J. F. Chris Landsea, J. Beven, Atlantic hurricane database (hurdat2) 1851-2014 (2014).